

DETAILED ACTION

This is a final action for application number 10/750,280 based on after non-final filed on 03/27/2008. The original application was filed on 12/30/2003. Claims 1 – 30 are currently pending and have been considered below. Claims 1, 7, 15, 21, and 25 are independent claims.

Applicant's Response

Applicant's arguments filed in the amendment filed 03/27/2008, have been fully considered but they are not persuasive. The reasons are set forth below.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

- Claims 1 - 6 and 25 - 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Matena et al. (US 7,302,609), and further in view of Brey et al. (US 2005/0108395).

Regarding claim 1, 25 Matena teaches a method comprising: establishing communication between a plurality of non-Java-based server nodes of a first instance and a plurality of Java-based server nodes of a second instance via an intermediate server, **[a flexible and extensible execution control system for distributed computer systems including multiple nodes interconnected through a network, (Abstract), wherein the multiple nodes are connected in a communication system via an intermediate server, These clustered servers are designed to allow applications to execute on multiple identically-configured J2EE server processes and perform transactions on data in a shared database, (Column 3, line 11)]**;

Matena et al. differs from the claimed invention is that generating a packet to be transmitted from one of the non-Java-based server nodes to one of the Java-based server nodes is not taught in Matena et al.

Brey et al. teaches generating a packet to be transmitted from one of the non-Java-based server nodes to one of the Java-based server nodes, **[A client node sends a packet to the server node, (Paragraph 0031)]**;

specifying in a header of the packet an address of a destination Java-based server node and information that indicates that the packet is generated by one of the non-Java-based server nodes, **[The access table is stored at the server node, and the table is sent in the packet as information, (Paragraph 0029)]**;

forwarding the packet to the intermediate server from the one of the non-Java-based server nodes, **[The server node receives the packet, STEP 602, and uses the access table to identify the allowable resources, (Paragraph 0032)];**

and forwarding the packet to the destination Java-based server node from the intermediate server based on the address provided in the header of the packet, **[The packet is rejected, Otherwise, the request is serviced, STEP 610. For example, data is written to or read from a particular device, (Paragraph 0032)].**

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to modify Matena by generating a packet to be transmitted from one of the non-Java-based server nodes to one of the Java-based server nodes as taught by Brey.

One of ordinary skill in the art would have been motivated to make this modification in order provide the advantage of generating a packet to be transmitted from one of the non-Java-based server nodes to one of the Java-based server nodes.

Regarding claim 2, 26 Brey et al. teaches that the method further comprising: generating a second packet to be transmitted from one of the Java-based server nodes to one of the non-Java-based server nodes, **[A client node sends a packet to the server node, (Paragraph 0031)];**

specifying in a header of the second packet an address of a destination non-Java-based server node and information that indicates that the packet is generated by

one of the Java-based server nodes, **[The access table is stored at the server node, and the table is sent in the packet as information, (Paragraph 0029)];**

forwarding the second packet to the intermediate server from the one of the Java-based server nodes, **[The server node receives the packet, STEP 602, and uses the access table to identify the allowable resources, (Paragraph 0032)];**

and forwarding the second packet to the destination non-Java-based server node from the intermediate server based on the address provided in the header of the second packet, **[The packet is rejected, Otherwise, the request is serviced, STEP 610. For example, data is written to or read from a particular device, (Paragraph 0032)].**

Regarding claim 3, 27 Matena teaches the method of claim 2, further comprising: maintaining a list of services performed by the non-Java-based server nodes, **[a service application controller 104, which is suitable for controlling applications that are services 107 including operating-system level processes, (Column 9, line 57)];**

and sending notification of a status of each of the listed services to the non-Java-based server nodes in the first instance, **[operations to obtain status information about the nodes, (Column 14, line 31)].**

Regarding claim 4, 28 Matena teaches that the method further comprising: maintaining a list of services performed by the Java-based server nodes, **[a service application controller 104, which is suitable for controlling applications that are services 107 including operating-system level processes, (Column 9, line 57)];**

and sending notification of a status of each of the listed services to the Java-based server nodes in the second instance, **[operations to obtain status information about the nodes, (Column 14, line 31)]**.

Regarding claim 5, 29 Matena teaches the method of claim 4, wherein the maintaining a list of services is accomplished by the intermediate server and the sending notification of a status of each of the listed services is accomplished by the intermediate server, **[Fig. 1, Ref # 103]**.

Regarding claim 6, 30 Matena teaches that the method further comprising: implementing Java 2 Platform Enterprise Edition (J2EE) applications in the Java-based server nodes, **[FIG. 30, Ref # J2EE]**.

- Claims 7 - 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Matena et al. (US 7,302,609), and further in view of Pedersen et al. (US 2003/0037148).

Regarding claim 7, Matena teaches a method comprising: establishing communication between a plurality of non-Java-based server nodes of a first instance and a plurality of Java-based server nodes of a second instance via an intermediate server.

Matena et al. differs from the claimed invention a first instance and a second instance are not taught in Matena et al.

Pedersen et al. teaches a system comprising: a first instance including a plurality of non-Java-based server nodes, each of the non-Java-based server nodes executing

software instructions to attach a header to a body of a packet, the header including information to specify that the packet originated from one of the non-Java-based server nodes, **[Fig. 2, Ref # 40, which is an instance with multiple nodes 34, 96, 24];**

a second instance including a plurality of Java-based server nodes, each of the Java-based server nodes executing software instructions to attach a header to a body of a packet, the header including information to specify that the packet originated from one of the Java-based server nodes, **[Fig. 2, Ref # 40, which is an instance with multiple nodes 34, 96, 24, wherein Instance 40 is a parameter handler that registers different instances];**

and a message server coupled between the first and second instances to establish communication there between, **[Fig. 2, Ref # 34, which is a server that sends a message to instance 40].**

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to modify Matena by including a first instance including a plurality of server nodes, each of the server nodes executing software instructions to attach a header to a body of a packet, the header including information to specify that the packet originated from one of the server nodes as taught by Pedersen.

One of ordinary skill in the art would have been motivated to make this modification in order provide the advantage of including a first instance including a plurality of server nodes, each of the server nodes executing software instructions to attach a header to a body of a packet, the header including information to specify that the packet originated from one of the server nodes.

Regarding claim 8, Pedersen et al. teaches the system of claim 7, wherein each of the instances further comprises a dispatcher to distribute client requests to the server nodes of the respective instance, **[Fig. 2, Ref # 30 which is an application requested by client 24, and the instance 40 is the distinct server node]**.

Regarding claim 9, Pedersen et al. teaches the system of claim 7, wherein the message server is to route message packets between the non-Java-based server nodes of the first instance and the Java-based server nodes of the second instance, **[Fig. 2, Ref # 54 and 58 wherein the 54 and 58 are messages from and to client node and instance 40]**.

Regarding claim 10, Pedersen et al. teaches the system of claim 7, wherein the message server is to assign a service identification associated with each type of services executed on the server nodes, **[Fig. 2, Ref # 80, which is a communication manager that moves the client identifier]**.

Regarding claim 11, Pedersen et al. teaches the system of claim 10, wherein the message server includes a service repository to maintain a list of the assigned service identification and corresponding service names, **[Fig. 2, Ref # 40, which maintains a table of addresses]**.

Regarding claim 12, Matena teaches that the method further comprising: maintaining a list of services performed by the Java-based server nodes, **[a service**

Art Unit: 2146

application controller 104, which is suitable for controlling applications that are services 107 including operating-system level processes, (Column 9, line 57)];

and sending notification of a status of each of the listed services to the Java-based server nodes in the second instance, **[operations to obtain status information about the nodes, (Column 14, line 31)].**

Regarding claim 13, Matena teaches the method of claim 4, wherein the maintaining a list of services is accomplished by the intermediate server and the sending notification of a status of each of the listed services is accomplished by the intermediate server, **[Fig. 1, Ref # 103].**

Regarding claim 14, Matena teaches that the method further comprising: implementing Java 2 Platform Enterprise Edition (J2EE) applications in the Java-based server nodes, **[FIG. 30, Ref # J2EE].**

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

- Claims 15 – 24 are rejected under 35 U.S.C. 102(e) as being anticipated

by Matena et al. (US 7,302,609).

Regarding claim 15 and 21, Matena teaches a message server comprising: a first communication interface to establish communication with a plurality of non-Java-based server nodes, **[Fig. 6, Ref # 605, which is one or more communication interfaces];**

a second communication interface to establish communication with a plurality of Java-based server nodes, **[Fig. 6, Ref # 605, which is one or more communication interfaces];**

and a controller to transfer packets between the non-Java-based server nodes and the Java-based server nodes, **[Fig. 1, Ref # 102, which is an execution controller]**.

Regarding claim 16, 22, Matena teaches the message server wherein the controller is to assign a service identification associated with each type of services executed on the server nodes, **[Fig. 1, Ref # 104, which identifies service such as Ref # 107]**.

Regarding claim 17, Matena teaches that the message server further comprises: a service repository maintains a list of the assigned service identification and corresponding service names, **[Fig. 11, Ref # Node Group]**.

Regarding claim 18, Matena teaches that the message server further comprises: a first repository to maintain a list of services currently being executed on the non-Java-based server nodes, **[Fig. 11, Ref # Node Group];**

and a second repository to maintain a list of services currently being executed on the Java-based server nodes, **[Fig. 11, Ref # Node Group]**.

Regarding claim 19, 20, 24 Matena teaches the method of claim 4, wherein the maintaining a list of services is accomplished by the intermediate server and the sending notification of a status of each of the listed services is accomplished by the intermediate server, **[Fig. 1, Ref # 103]**.

Regarding claim 23 Matena teaches that the method further comprising: maintaining a list of services performed by the Java-based server nodes, **[a service application controller 104, which is suitable for controlling applications that are services 107 including operating-system level processes, (Column 9, line 57)]**;

and sending notification of a status of each of the listed services to the Java-based server nodes in the second instance, **[operations to obtain status information about the nodes, (Column 14, line 31)]**.

Response to Arguments

The Applicant Argues:

That Matena does not teach a controller to transfer packets between the non-Java based server nodes and the Java-based server nodes.

In response, the examiner respectfully submits: Matena teaches a controller to transfer packets between non-Java based server nodes and Java-based server nodes, and this controller called a Java Application controller shown in Fig. 36, wherein The Java Application Controller ("JAC") is another exemplary application controller used in the preferred embodiment of the invention. Although the description of JAC focuses on the execution control of Java applications, the system and method of the invention equally apply to any distributed applications written in any programming language if they have a similar application execution model to Java applications, (**Matena et al., Col. 36, Lines 52-64**).

The Applicant Argues:

That Matena does not teach that packets are being transferred between non-Java based server nodes and a Java-based server nodes.

In response, the examiner respectfully submits: Matena teaches that packets are being transferred between non-Java based server nodes and a Java-based server nodes as shown in Fig. 7 of Matena, wherein A node controller 701 implements the following functions: node membership 702, master election 703, start process 704, stop process 705, and detect process failure 706. The "membership" function 702 includes sending messages to other nodes and receiving messages from other nodes in order to determine which nodes are currently participating in the distributed computer system, (**Matena et al., Col. 11, Lines 52-60**).

The Applicant Argues:

That Matena does not teach a communication between a plurality of non-Java based server nodes and a plurality of Java-based server nodes.

In response, the examiner respectfully submits: Matena teaches a communication between a plurality of non-Java based server nodes and a plurality of Java-based server nodes as shown in Fig. 2 wherein a distributed computer system including nodes interconnected by a network. The distributed computer system includes an execution control system. The exemplary distributed computer system includes six nodes, (Matena et al., Col. 10, Lines 3-10), wherein each node has a communication interface to communicate with other nodes as shown in Fig. 6, Ref # 605 and the node is a computer system including one or more central-processing units (CPU) 602, main memory 603, optional secondary storage 604, and one or more communication interfaces 605. The communication interfaces 605 allow a node 601 to communicate with other nodes and optionally with other computers over a network 606, (Matena et al., Col. 11, Lines 38-45).

The Applicant Argues:

That Brey does not teach generating a packet to be transmitted from one of the non-Java-based server nodes to one of the Java-based server nodes.

In response, the examiner respectfully submits: Brey teaches generating a packet to be transmitted from one of the non-Java-based server nodes to one of the Java-based server nodes as shown in Fig. 6, wherein the client generates a packet

request and sends it to a node, wherein specifying in a header of the packet an address of a destination as shown in Fig. 4 wherein This includes, for instance, receiving an indication of the resource assignments from a system administrator or other entity, and/or using a technique to decide which resources are to be allocated to which node, **(Brey et al., Paragraph 28, Page 2)**.

The Applicant Argues:

That Pederson does not teach a first and a second instance which include a plurality of server nodes.

In response, the examiner respectfully submits: Pederson teaches a first and a second instance which include a plurality of server nodes as shown in Fig. 2 of Ref # 40, wherein the instance includes a plurality of server nodes 24,34, and 96, wherein and referring to FIG. 2, the client process 56 on client node 24 makes a request 54 to the network master information node 40 to obtain the address of a server node 34 which includes the desired application 62, **(Pederson et al., Paragraph 20, Page 2)**.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to **Shaq Taha** whose telephone number is 571-270-1921. The examiner can normally be reached on 8:30am-5pm Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, **Jeff Pwu** can be reached on 571-272-6798.

Art Unit: 2146

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/S. T./

Examiner, Art Unit 2146

/Jeffrey Pwu/

Supervisory Patent Examiner, Art Unit 2146